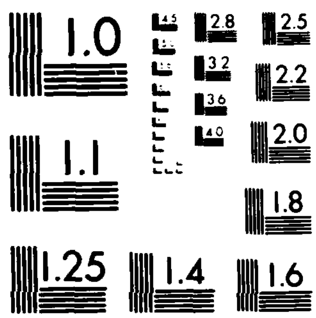
AD-A138 541 DATA COMPRESSION(U) STANFORD UNIV CA INFORMATION SYSTEMS LAB R M GRAY 21 FEB 84 ARO-17372.7-EL DAAG29-80-K-0073 1/1

UNCLASSIFIED F/G 9/2 NL

END
DATE FILMED
4 - 84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD A138541

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ARO 17372.7-EL	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): Data Compression		5. TYPE OF REPORT & PERIOD COVERED: Final Report 1 Sep 1980 - 31 Jan 1984
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Robert M. Gray		8. CONTRACT OR GRANT NUMBER(s): DAAG29-80-K-0073
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Stanford Univ		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: N/A
11. CONTROLLING OFFICE NAME AND ADDRESS: U. S. Army Research Office, Post Office Box 12211, Research Triangle Park, NC 27709		12. REPORT DATE: 1984
		13. NUMBER OF PAGES: 21
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report): Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

ELECTE MAR 5 1984

A

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Aided Design
Data Compression
Algorithms

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The fundamental goal of this contract was to develop computer aided design algorithms for data compression systems and to study the performance and complexity of these systems via simulation and mathematical analysis.

over

84 03 02 043

DTIC FILE COPY

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

ARO 17372.7-EL Continued

20. ABSTRACT CONTINUED

Since distortion is inevitable in compression systems, a design goal is to minimize the average distortion for a given communication or storage capacity or, equivalently, to minimize the communication or storage capacity subject to satisfactory data fidelity.

This project was devoted to finding design algorithms which begin with a code of a fixed structure and then iteratively improve the code in the sense of producing codes with lower distortion and hence better fidelity. The code structures are chosen to be implementable using current technology. The basic structure of all of the systems developed is well suited to VLSI implementation: a minimum distortion search algorithm on a chip communicating with off-board storage for codebooks and next-state transition functions.

As most of the systems developed and studied as part of this contract are described in detail in published papers, in papers currently being considered for publication, or in papers in preparation, this final report presents only a brief survey of the accomplishments under the contract toge'her with citations of the papers where the detailed development and results may be found.

A1

DATA COMPRESSION

FINAL TECHNICAL REPORT

Robert M. Gray
Principal Investigator

21 February 1984

U.S. ARMY RESEARCH OFFICE

Contract Number DAAG29-80-K-0073

Information Systems Laboratory
Stanford University

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

Contents

Introduction and Summary

The fundamental goal of this contract was to develop computer aided design algorithms for data compression systems and to study the performance and complexity of these systems via simulation and mathematical analysis. Data compression is the reduction of analog or high rate digital data to relatively low rate digital information. Compression is desirable in order to minimize communication channel capacity requirements in a fixed rate communication system, to minimize packet size or transmission time in a packet or burst communication system, or to minimize digital memory storage requirements in systems where the data is stored for future reproduction, e.g., taped satellite data or synthesized speech in talking computers. Since distortion is inevitable in compression systems, a design goal is to minimize the average distortion for a given communication or storage capacity or, equivalently, to minimize the communication or storage capacity subject to satisfactory data fidelity.

This project was devoted to finding design algorithms which begin with a code of a fixed structure and then iteratively improve the code in the sense of producing codes with lower distortion and hence better fidelity. The code structures are chosen to be implementable using current technology. The basic structure of all of the systems developed is well suited to VLSI implementation: a minimum distortion search algorithm on a chip communicating with off-board storage for codebooks and next-state transition functions. As new and better design algorithms are developed, the chips can be updated by simply reburning the codebook and transition ROM's.

The original proposal emphasized the application of techniques developed at Stanford for the design of vector quantizers to other data compression systems--trellis encoding systems and hybrid vector quantization/tree encoding systems in particular. Success on these code structures led to the development of design algorithms for other code structures: finite state vector quantizers, predictive vector quantizers, feedback vector quantizers, gain/shape vector quantizers, and adaptive vector quantizers. The initial focus on coding Gaussian random processes, speech waveforms, and linear predictive coded (LPC) speech parameter vectors was expanded to include image coding applications.

As most of the systems developed and studied as part of this contract are described in detail in published papers, in papers currently being considered for publication, or in papers in preparation, this final report presents only a brief survey of the accomplishments under the contract together with citations of the papers where the detailed development and results may be found. Copies of reprints of papers published in journals will be forwarded to ARO as they become available. A complete summary of all of the work supported by this project except for the speech recognition work may be found in [1], a preprint of which has already been forwarded to ARO.

The success of several of the techniques developed under the project is attested to by their application to problems of speech and image coding and speech recognition by a variety of organizations, including the U.S. Naval Research Laboratory, Bell Laboratories, IBM, Matsushita, and NTT Musashino Research Laboratory. Active research on applications of these techniques is also currently under way at numerous universities, including the University of California at Berkeley and at Santa Barbara, the University of Mexico, Osaka University, Ehime University, Insitutut fur Angewandte Physik der Johann-Wolfgang-Goethe Universitat in Frankfurt, Germany, and the California State University, San Diego. The bulk of the current research is now being conducted in Japan, where devices based on design techniques developed under this project are now in development.

Memoryless Vector Quantization and Data Compression

Mathematically, a k-dimensional memoryless vector quantizer or, simply, a VQ (without modifying adjectives) consists of two mappings: an encoder γ which assigns to each input vector $\mathbf{x}=(x_0,x_1, \cdots ,x_{k-1})$ a channel symbol $\gamma(\mathbf{x})$ in some channel symbol set $\mathbf{M}$, and a decoder β assigning to each channel symbol u in $\mathbf{M}$ a value in a reproduction alphabet $\hat{\mathbf{A}}$. The channel symbol set is often assumed to be a space of binary vectors for convenience, e.g., $\mathbf{M}$ may be the set of all 2^R binary R-dimensional vectors. The reproduction alphabet may or may not be the same as the input vector space; in particular, it may consist of real vectors of a different dimension.

If $\mathbf{M}$ has M elements, then the quantity $R = \log_2 M$ is called the *rate* of the quantizer in bits per vector and $r = R/k$ is the rate in bits per symbol or, when the input is a sampled waveform, bits per sample.

The application of a quantizer to data compression is depicted in Figure 1. The input data vectors might be consecutive samples of a waveform, consecutive parameter vectors in a voice coding system, or consecutive rasters or subrasters in an image coding system. For integer values of R it is useful to think of the channel symbols, the encoded input vectors, as binary R-dimensional vectors. As is commonly done in information and communication theory, we assume that the channel is noiseless, that is, that $U_n = \hat{U}_n$. While real channels are rarely noiseless, the joint source and channel coding theorem of information theory implies that a good data compression system designed for a noiseless channel can be combined with a good error correction coding system for a noisy channel in order to produce a complete system. In other words, the assumption of a noiseless channel is made simply to focus on the problem of data compression system design and not to reflect any practical model.

The goal of such a quantization system is to produce the "best" possible reproduction sequence for a given rate R. To quantify this idea, to define the performance of a quantizer, and to complete the definition of a quantizer requires the idea of a distortion measure: A distortion measure d is an assignment of a cost $d(\mathbf{x},\hat{\mathbf{x}})$ of reproducing any input vector $\mathbf{x}$ as a reproduction vector $\hat{\mathbf{x}}$. Given such a distortion measure, we can quantify the performance of a system by an average distortion $Ed(\mathbf{X},\hat{\mathbf{X}})$ between the input and the final reproduction: A system will be good if it yields a small average distortion. In practice, the important average is the long term sample average or time average

$$\lim_{n \to \infty} \frac{1}{n} \sum_{i=0}^{n-1} d(\mathbf{X}_i,\hat{\mathbf{X}}_i) ,$$

provided, of course, that the limit makes sense. For example if the process is stationary and ergodic, then with probability one the above limit exists and equals an expectation $E(d(\mathbf{X},\hat{\mathbf{X}}))$. We here assume that such conditions are met. General conditions for this assumption to be valid have been developed [2].

Ideally a distortion measure should be tractable to permit analysis, computable so that it can be evaluated in real time and used in minimum distortion systems, and subjectively meaningful so that large or small quantitative distortion measures correlate with bad and good subjective quality. We do not consider the difficult and controversial issues of selecting a distortion measure; we assume that one has been selected and consider means of designing systems which yield small average distortion. While several distortion measures have been considered, two have received the most attention because of their popularity and simplicity: The squared error distortion measure and the Itakura-Saito (IS) distortion. The squared error distortion measure and its weighted generalizations is useful for waveform coding applications since minimizing its average is equivalent to minimizing the power in the reproduction error signal, possibly with selective frequency weighting or weighting based on long term input power. The IS distortion measure is useful in voice coding applications where the receiver is sent a linear model of the underlying voice production process. More generally, this distortion measure is a special case of a minimum relative entropy or discrimination measure and VQ using such distortion measures can be viewed as an application of the minimum relative entropy pattern classification technique introduced by Kullback as an application of information theory to statistical pattern classification. This latter connection suggests that the distortion measure may also be useful in recognition and classification applications. Details of the definition and properties of this distortion measure (which require LPC notation and jargon) may be found in [3, 4, 5, 6].

For this summary, we note simply that this is the distortion measure implicitly minimized by LPC speech systems, the best quality very low rate digital speech systems, and that the distortion measure is relatively complicated--it is not a simple function of an error vector, it is not symmetric in its input and output arguments, and it is not a metric or distance.

A VQ is said to be optimal if it minimizes an average distortion $Ed(\mathbf{X},\beta(\gamma(\mathbf{X})))$. A general algorithm for the design of vector quantizers that are at least locally optimal was developed by generalizing a technique of Lloyd for the design of optimal PCM systems [7, 8]. The algorithm begins with an initial code and then iteratively optimizes the encoder for the decoder and vice versa in the sense of reducing the long term average distortion for a training sequence of data typical of the source to be compressed. Before the beginning of this contract, the basic algorithm had been developed for memoryless vector quantizers and a variety of initialization schemes for the algorithm had been developed. The technique was used successfully on speech waveforms, LPC speech parameter vectors, and a variety of random process models.

Variations of Memoryless Vector Quantizers

Before considering vector quantizers with memory, we consider two important variations of memoryless VQ developed in this project. While mathematically suboptimal, both variations yield efficient implementations that can provide equal performance and rate with smaller computational complexity. Codes can be designed for all of these structures using variations of the basic design algorithm.

Tree-Searched VQ

Tree-searched vector quantizers were first proposed by Buzo *et al.* [3]. They can be viewed as a vector generalization of a successive approximation scalar quantizer. The code has a tree structure and each input vector is encoded using a sequence of small, e.g., binary choices rather than a single search of a full codebook. The encoding is not optimal and the memory is increased, but in some applications the coding is nearly optimal. The search complexity is, however, greatly reduced.

Gain/Shape VQ

A gain/shape VQ is an example of a product/multistep VQ where separate attributes of the input vector are encoded using separate, but interdependent, codebooks. In a gain/shape VQ separate codes are used to code the "shape" and "gain" of the waveform, where the "shape" is defined as the original input vector normalized by removal of a "gain" term such as energy in a waveform coder or or LPC residual energy in a vocoder. Gain/shape encoders were introduced by Buzo *et al.* [3] and were subsequently extended and optimized by Sabin and Gray [9, 10]. The basic idea is to use VQ only on the complicated shape vector, and then use a simple scalar code, which is dependent on the shape codeword selected, to encode the gain. This permits higher rates and hence better quality w' . reasonable memory and computation requirements. Such systems have a much wider dynamic range than ordinary VQ.

Separating Mean VQ

Another example of a product/multistep code is the separating mean VQ where a sample mean instead of an energy term is removed [11]. In a separated mean VQ one first uses a scalar quantizer to code the sample mean of a vector, then the coded sample mean is subtracted from all of the components of the input vector to form a new vector with approximately zero sample mean. This new vector is then vector quantized. The basic motivation here is that in image coding the sample mean of pixel intensities in a small rectangular block represents a relatively slowly varying average background value of pixel intensity around which there are variations.

Feedback Vector Quantizers

Memory can be incorporated into a vector quantizer in a simple manner by using different codebooks for each input vector, where the codebooks are chosen based on past input vectors. The decoder must know which codebook is being used by the encoder in order to decode the channel symbols. This can be accomplished in two ways: 1) The encoder can use a codebook selection procedure that depends only on past encoder outputs and hence the codebook sequence can be tracked by the decoder. 2) The decoder is informed of the selected codebook via a special low-rate side channel. The first approach is called feedback vector quantization and is the topic of this section. The name follows because the encoder output is "fed back" for use in selecting the new codebook. A feedback vector quantizer can be viewed as the vector extension of a scalar adaptive quantizer with backward estimation (AQB). The second approach is the vector extension of a scalar adaptive quantizer with forward estimation (AQF) and is called simply adaptive vector quantization. Observe that systems can combine the two techniques and use both feedback and side information. We also point out that unlike most scalar AQB and AQF systems, the vector analogs considered here involve no explicit estimation of the underlying densities.

It should be emphasized that the results of information theory imply that VQ's with memory can do no better than memoryless VQ's in the sense of minimizing average distortion for a given rate constraint. In fact, the basic mathematical model for a data compression system in information theory is exactly a memoryless VQ and such codes can perform arbitrarily close to the optimal performance achievable using any data compression system. The exponential growth of computation and memory with rate, however, may result in nonimplementable VQ's. A VQ with memory may yield the desired distortion with practicable complexity.

A general feedback VQ can be described as follows Suppose now that we have a space **S** whose members we shall call states and that for each state s in **S** we have a separate quantizer: an encoder γ_s and a decoder β_s. The channel codeword space **M** is assumed to be the same for all of the VQ's. Consider a data compression system consisting of a sequential machine such that if the machine is in state s, then it uses the quantizer with encoder γ_s and decoder β_s. It then selects its next state by a mapping called a next-state function or state-transition function f such that given a state s and a channel symbol u, then $f(u,s)$ is the new state of the machine. More precisely, given a sequence of input vectors $\{\mathbf{x}_n;\ n=0,1,2,...\}$ and an initial state s_0, then the subsequent state sequence s_n, channel symbol sequence u_n, and reproduction sequence $\hat{\mathbf{x}}_n$ are defined recursively for $n=0,1,2,...$ as

$$u_n = \gamma_{s_n}(\mathbf{x}_n)\ ,\ \hat{\mathbf{x}}_n = \beta_{s_n}(u_n)\ ,\ s_{n+1} = f(u_n,s_n)\ .$$

Since the next state depends only on the current state and the channel codeword, the decoder can track the state if it knows the initial state and the channel sequence. The freedom to use different quantizers based on the past without increasing the rate should permit the code to perform better than a memoryless quantizer of the same dimension and rate.

If the state space is finite, then the resulting system is called a finite-state vector quantizer or FSVQ. For an FSVQ, all of the codebooks and the next-state transition table can all be stored in ROM, making the general FSVQ structure amenable to LSI or VLSI implementation [12].

Observe that a memoryless vector quantizer can be modeled as a feedback vector quantizer or finite-state vector quantizer with only a single state.

Three design algorithms for feedback vector quantizers using variations on the generalized Lloyd algorithm were studied as part of this project.

i) Vector Predictive Quantization

Cuperman and Gersho [13, 14] proposed a vector predictive coder or vector predictive quantizer (VPQ) which is a vector generalization of DPCM or predictive quantization. For a fixed predictor, the VQ design algorithm is used to design a VQ for the prediction error sequence. Cuperman and Gersho considered several variations on the basic algorithm, some of which will be later mentioned.

Chang and Gray [15, 1] developed an extension to Cuperman and Gersho's algorithm which begins with their system and then uses a stochastic gradient algorithm to iteratively improve the vector linear predictor coefficients, that is, to better match the predictor to the quantizer. A stochastic gradient algorithm is also used to improve the resulting codebooks.

ii) Product/Multistep FVQ

A second basic approach for designing feedback vector quantizers which is quite simple and works quite well is to use a product multistep VQ such as the gain/shape VQ or the separating mean VQ and use a simple feedback quantizer on the scalar portion and an ordinary memoryless VQ on the remaining vector. This approach was developed in [16] for gain/shape VQ of LPC parameters and in [11] for separating mean VQ of images. Both efforts used simple scalar predictive quantization for the feedback quantization of the scalar terms.

iii) Finite State Vector Quantizers

The first general design technique for finite-state vector quantizers was reported by Foster and Gray [17, 18], and developed further developed in [19]. There are two principal design components: 1. Design an initial set of state codebooks and a next-state function using an *ad hoc* algorithm. 2. Given the next-state function, use a variation of the basic algorithm to improve the state codebooks. The second component is accomplished by a slight extension of the basic algorithm that is similar to the extension of [20] for the design of trellis encoders. The best design algorithm found for the first step is called the omniscient state design technique and it involves the design of an idealized state sequence for which the ordinary VQ design algorithm can be applied to the separate sub-training sequences associated with each state. This idealized state is then approximated by a trackable state selection based on encoder outputs. The state sequences of such codes can be viewed as a form of coarse prediction of the next input vector. A design algorithm similar to the omniscient design technique was independently developed by Haoui and Messerschmitt [21].

After the basic design algorithms were developed, techniques based on the theory of adaptive stochastic automata were applied to iteratively improve the transition structure of the finite state machines used for compression. These algorithms can be viewed as a prescription for a computer to efficiently modify the parameters of a coding system while viewing the quality of the output in order to obtain the best possible average quality [22]

Tree and Trellis Encoders

The actions of the decoder of a feedback VQ can be depicted as a directed graph or tree. Instead of using the ordinary VQ encoder which is only permitted to look at the current input vector in order to decide on a channel symbol, one could use a algorithms such as the Viterbi algorithm, M-algorithm or $M.L$-algorithm, Fano algorithm, or stack algorithm for a minimum cost search through a directed graph and search several levels ahead into the tree or trellis before choosing a channel symbol. This introduces an additional delay into the encoding of several vectors, but it ensures better long run average distortion behavior. This technique is called tree or trellis encoding and is also referred to as look-ahead coding, delayed decision coding, and multipath search coding. [20]

A natural variation of the basic algorithm for designing FSVQ's can be used to design trellis encoding systems where the vector quantizer encoder which finds the minimum distortion reproduction for a single input vector is replaced by a Viterbi or other search algorithm which searches the decoder trellis to some fixed depth to find a good long term minimum distortion path. Scalar and simple two dimensional vector trellis encoding systems were designed in [20] using this approach.

Trellis encoding systems are not really vector quantization systems as we have defined them since the encoder is permitted to search ahead to determine the effect on the decoder output of several input vectors while a vector quantizer is restricted to search only a single vector ahead. The two systems are intimately related, however, and a trellis encoder can always be used to improve the performance of a feedback vector quantizer. Very little work has yet been done on vector trellis encoding systems.

Adaptive Vector Quantization

As a final class of VQ we consider systems that use one VQ to adapt a waveform coder, which might be another VQ. The adaptation information is communicated to the receiver via a low rate side information channel.

The various forms of vector quantization using the Itakura-Saito family of distortion measures can be considered as model classifiers, that is, they fit an all-pole model to an observed sequence of sampled speech. When used alone in an LPC VQ system, the model is used to synthesize the speech at the receiver. Alternatively, one could use the model selected to choose a waveform coder designed to be good for sampled waveforms that produce that model. For example, analogous to the omniscient design of FSVQ one could design separate VQ's for the subsequences of the training sequence encoding into common models. Both the model index and the waveform coding index are then sent to the receiver. Thus LPC VQ can be used to adapt a waveform coder, possibly also a VQ or related system. This will yield a system typically of much higher rate than the LPC VQ system, but potentially of much better quality since the codebooks can be matched to local behavior of the data. The model VQ typically operates on a much larger vector of samples and at a much lower rate in bits per sample than does the waveform coder and hence the bits spent on specifying the model through the side channel are typically much fewer than those devoted to the waveform coder.

There are a variety of such possible systems since both the model quantizer and the waveform quantizer can take on many of the structures so far considered. One example was developed for this project by Chang and Gray [15, 1]. The system uses an ordinary LPC VQ as the classifier and with a stochastic gradient algorithm run on each of the vector predictive quantizers in order to improve the prediction coefficients for the corresponding codebooks.

A different system using LPC VQ for adaptation and a trellis waveform encoder was developed by [20]. Both of these systems used the basic algorithm to design both the model VQ and the waveform coders.

Many other variations on the general theme are possible and the structure is a promising one for processes such as images and speech that exhibit local stationarity, that is, slowly varying short term statistical behavior. The use of one VQ to partition a training sequence in order to design good codes for the resulting distinct subsequences is an intuitive approach to the computer-aided design of adaptive data compression systems.

References

1. R.M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, (April 1984). to appear.
2. M.J. Sabin and R.M. Gray, *Global convergence and empirical consisitency of the generalized Lloyd algorithm*, Submitted for publication 1984.
3. A. Buzo, A.H. Gray Jr., and R.M. Gray and J.D. Markel, "Speech coding based upon vector quantization," *IEEE Transactions on Acoustics Speech and Signal Processing* **ASSP-28** pp. 562-574 (October 1980).
4. R.M.Gray, A. Buzo, A.H. Gray Jr., and Y. Matsuyama, "Distortion measures for speech processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* **ASSP-28,** pp. 367-376 (August 1980).
5. R.M. Gray, A.H. Gray Jr., G. Rebolledo, and J.E. Shore, "Rate distortion speech coding with a minimum discrimination information distortion measure," *IEEE Transactions on Information Theory* **IT-27**(6) pp. 708-721 (Nov. 1981).
6. J.E. Shore and R.M. Gray, "Minimum-cross-entropy pattern classification and cluster analysis," *IEEE Transactions on Pattern Analysis and Machine Intellegence* **PAMI-4** pp. 11-17 (Jan. 1982).
7. Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications* **COM-28** pp. 84-95 (January 1980).
8. S.P.Lloyd, *Least squares quantization in PCM,* Published in the March 1982 special issue on quantization of the September 1957.
9. M.J. Sabin and R.M. Gray, "Product code vector quantizers for speech waveform coding," *Conference Record Globecom '82,* pp. 1087-1091 (December 1982).
10. M.J. Sabin and R.M. Gray, "Product code vector quantizers for waveform and voice coding," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* (April 1984). to appear.
11. R.L. Baker and R.M. Gray, "Differential vector quantization of achromatic imagery," *Proceedings of the International Picture Coding Symposium,* (March 1983).
12. J. Foster, J. Newkirk, and R.M. Gray, *VLSI implementation of a finite-state vector quantization waveform encoder,* submitted for publication 1983.
13. V. Cuperman and A. Gersho, "Adaptive differential vector coding of speech," *Conference Record, GlobeCom 82,* pp. 1092-1096 (December 1982).
14. V. Cuperman and A. Gersho, *Vector predictive coding of speech at 16 Kb/s,* Submitted for possible publication 1983.
15. P.C. Chang, Ph. D. Research 1983.
16. D. Wong, B.-H. Juang, and A.H. Gray Jr., "An 800 bit/s vector quantization LPC vocoder," *IEEE Transactions on Acoustics Speech and Signal Processing* **ASSP-30** pp. 770-779 (October 1982).
17. J. Foster and R.M. Gray, "Finite-state vector quantization," *Abstracts of the 1982 IEEE International Symposium on Information Theory,* (June 1982).
18. J. Foster, *Finite-state vector quantization for waveform coding,* Ph. D. Dissertation November 1982.

19. J. Foster , R.M. Gray, and M. Ostendorf, "Finite-state vector quantization for waveform coding," *IEEE Transactions on Information Theory*, (1984). to appear.

20. L.C. Stewart, R.M. Gray, and Y. Linde, "The design of trellis waveform coders," *IEEE Transactions on Communications* **COM-30** pp. 702-710 (April 1982).

21. A. Haoui and D.G. Messerschmitt, "Predictive vector quantization," *Proceedings ICASSP*, (1984).

22. M. Ostendorf and R.M. Gray, *An algorithm for the design of labeled-transition finite-state vector quantizers*, submitted for publication 1984.

Contributions

Given the preceding general descriptions, we can now summarize the contributions of this project. This section lists all of the papers published with the full or partial support of this contract.

The initial contributions were the extension of the original VQ design algorithms to design trellis encoding systems for Gaussian processes and for speech waveforms. These techniques were combined with LPC VQ techniques to obtain an adaptive midrange speech compression system that yielded good quality speech at 1 bit per sample with lower complexity than competing APC schemes [1].

Another early contribution was the study of the performance and complexity tradeoffs for full search VQ and tree-searched VQ applied to Gauss Markov sources [2].

The basic VQ design techniques were applied to image coding to obtain good quality images at rates of 1/2 to 1 bit per pixel [3]. In order to improve implementation efficiency and to better handle dynamic range, gain/shape VQ and separating mean VQ were developed, the first being used primarily for speech waveforms and LPC parameter compression [4,5] and the second for image coding applications [6].

The basic algorithms for designing finite state vector quantizers were developed for this project and applied to Gaussian processes and speech waveforms [7,8] and LPC parameter vectors [9].

Another feedback quantizer, the separating-mean FVQ was developed [6] and successfully used for image coding applications at rates of 1 bit per pixel and less. More detailed papers on the image coding applications are currently in preparation.

A variety of predictive vector quantizers and adaptive vector quantizers have been developed and preliminary results have been obtained by Chang and Gray [10, 11], but work is not yet complete. We are attempting to find funding to continue this work.

Recently VQ has also been successfully used in isolated word recognition systems without dynamic time warping by using either separate codebooks for each utterance or by mapping trajectories through one or more codebooks. We have also developed initial results along these lines including and endpoint algorithm suitable for use with VQ-based compression and recognition systems [12] and a simple vowel recognition system [13]. We believe that this too is a promising area and we are seeking additional funds from private industry to continue the project.

Publications Supported by the Project

References

1. L.C. Stewart, R.M. Gray, and Y. Linde, "The design of trellis waveform coders," *IEEE Transactions on Communications* **COM-30** pp. 702-710 (April 1982).
2. R. M. Gray and Y. Linde, "Vector quantizers and predictive quantizers for Gauss-Markov sources," *IEEE Transactions on Communications* **COM-30** pp. 381 - 389 (Feb. 1982).
3. R.L. Baker and R.M. Gray, "Image compression using non-adaptive spatial vector quantization," *Conference Record of the Sixteenth Asilomar Conference on Circuits Systems and Computers*, (October 1982).
4. M.J. Sabin and R.M. Gray, "Product code vector quantizers for speech waveform coding," *Conference Record Globecom '82*, pp. 1087-1091 (December 1982).
5. M.J. Sabin and R.M. Gray, "Product code vector quantizers for waveform and voice coding," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, (April 1984). to appear.
6. R.L. Baker and R.M. Gray, "Differential vector quantization of achromatic imagery," *Proceedings of the International Picture Coding Symposium*, (March 1983).
7. J. Foster and R.M. Gray, "Finite-state vector quantization," *Abstracts of the 1982 IEEE International Symposium on Information Theory*, (June 1982).
8. J. Foster , R.M. Gray, and M. Ostendorf, "Finite-state vector quantization for waveform coding," *IEEE Transactions on Information Theory*, (1984). to appear.
9. M. Ostendorf and R.M. Gray, *An algorithm for the design of labeled-transition finite-state vector quantizers*, submitted for publication 1984.
10. P.C. Chang, Ph. D. Research 1983.
11. R.M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, (April 1984). to appear.
12. C. Tsao and R.M. Gray, "An endpoint detector for LPC speech using residual error look-ahead," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, (March 1984).
13. C. Tsao and R.M. Gray, "An approach to speaker-dependent vowel recognition using vector quantization," *Conference Record, Towards Robustness in Speech Recognition*, Speech Science Publications, Santa Barbara, CA, (November 1983).

Participating Scientific Personnel and Degrees Awarded

Robert M. Gray, Principal Investigator

Students:

L.C. Stewart, earned Ph.D. on project, June 1981

John Foster, earned Ph.D. on project, November 1982 (Foster's salary was paid for by a Bell Labs Ph.D. fellowship, but he was an active participant in the project)

R.L. Baker, Ph.D. expected spring 1984

M.J. Sabin, Ph.D. expected spring 1984

M. Ostendorf, Ph.D. expected summer 1984

C. Tsao (Salary paid for by a fellowship from the government of Singapore, *but an active participant in the project)*

P.C. Chang